View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Scholarly Materials And Research @ Georgia Tech

EFFICIENT LABELING TECHNIQUE AND INTERPRETABLE DEEP NEURAL NETWORK FOR THE CLASSIFICATION OF SEIZURES USING CONTINUOUS ELECTROENCEPHALOGRAMS

A Thesis
Presented to
The Academic Faculty

By

Olivier Deiss

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in the
School of Computer Science

Georgia Institute of Technology

May 2018

Copyright © Olivier Deiss 2018

EFFICIENT LABELING TECHNIQUE AND INTERPRETABLE DEEP NEURAL NETWORK FOR THE CLASSIFICATION OF SEIZURES USING CONTINUOUS ELECTROENCEPHALOGRAMS

Approved by:

Dr. Jimeng Sun, Advisor
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. M. Brandon Westover
Department of Neurology
Massachusetts General Hospital

Dr. James M. Rehg
School of Interactive Computing
Georgia Institute of Technology

Date Approved: April 23, 2018

To Maman, Mimi, Mouche, Big and Billie

ACKNOWLEDGEMENTS

At Georgia Tech, I am first thankful to my advisor, Professor Jimeng Sun, for welcoming me to be part of his dynamic research team, and for having been a great mentor over the past year, providing guidance and support all along. Thank you Siddharth Biswal for insights, talks and support, and to Jeffrey Valdez for your coordinating work in the lab. I would also like to thank Professor James Rehg for his guidance on my work and helpful reviews.

I have been closely working with great minds at Massachusetts General Hospital that I would also like to thank. This includes Doctor M. Brandon Westover, for making it possible to work on such meaningful studies and for his great insights and hard work that tremendously helped, as well as Haoqi Sun and Jing Jin together for their work.

I finally thank my parents for giving me the best education one can wish, Aurélie for always being here, laughing in any situation, and Misha for almost always supporting my passion for computer science, this work, and personal projects that took almost all of my free time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis focuses on the classification of seizures, together with finding efficient and scalable ways to obtain high-quality datasets in order to train deep neural networks. It was motivated by the need to automate the classification of seizure patterns. In fact, roughly 30% of critically ill patients in ICUs suffer seizures or related patterns of harmful electrical activity of the brain. While seizures do damage the brain, most seizures in ICU patients occur without any obvious or overt clinical signs, and are thus detectable only by continuous electroencephalography (cEEG). cEEGs are recordings of the brain activity, often lasting over several hours.

Manually labeling all the recordings to detect such patterns is infeasible, and the problem is a great candidate for the application of automatic classifiers. In particular, deep neural networks are promising, as they already perform well in a wide range of other tasks. However, the key to obtaining robust classifiers is an efficient label acquisition process. Data labeling is often challenging and subject to high levels of label noise. This can arise even when classification targets are well defined, for example if instances to be labeled are more difficult than the prototypes used to define the class. This leads to disagreements among the expert community and leaves room for mis-interpretation of the concepts.

Therefore, although cEEG monitoring yields large volumes of data, labeling costs and difficulty make it hard to build a classifier. While experts agree on the labels of clear-cut examples of cEEG patterns, labeling many real-world cEEG data can be extremely challenging. Thus, a large number of sequences might be mislabeled, making training accurate deep learning models a really challenging task.

This work explores ways to efficiently scale the labeling efforts in an environment where manual annotation is error-prone due to the complexity of the task, concurrently with the design of an interpretable model, suitable for medical use. One of the results include a method for human and machine co-learning, where experts become consistent in

the labeling task, allowing to improve the quality of the dataset, while the model becomes stronger at correctly classifying inputs in the right category of seizure. This method is called HAMLET: a novel Human And Machine co-LEarning Technique. Using this system, it is possible to obtain a dataset that is suitable for training of deep learning models on challenging tasks, like the classification of seizures based on continuous EEG recordings. The core of the system integrates the constraint that some sample points cannot be reliably labeled even by human experts. In brief, during training, HAMLET is allowed to challenge the decision of human experts regarding the labels of certain difficult cases.

CHAPTER 1
INTRODUCTION

Over the last decade, thanks to the large-scale adoption of Electronic Health Records throughout the United States, the field of Health Informatics has gathered momentum, allowing increased collaboration between healthcare providers and the use of technology for improved patient care. The data-driven future of medicine looks promising. So far, numerous projects and successful systems have already been deployed and assist experts in taking care of patients. New standards and specifications are built to facilitate the exchange of medical data. SMART (Substitutable Medical Applications, Reusable Technologies) is a standard framework that allows the development of healthcare applications available at any institution. Similar in spirit, FHIR (Fast Healthcare Interoperability Resources) is being widely adopted and will make it even easier to securely and efficiently exchange data between experts and research groups in the medical field (D. Bender et al., 2013 [1]). Both projects now currently work in conjunction, FHIR focusing on the exchange standard and SMART formalizing how applications should interact with FHIR, in a framework called SMART-on-FHIR (J. C. Mendel et al., 2016 [2]). Health Informatics is even embracing open-source, with incentives like HAPI-FHIR, aiming to provide an open implementation of the FHIR standard.

Concurrently, for a wide spectrum of real-world applications, ranging from image classification (K. He et al., 2015 [3]), face and speech recognition (A. Y. Hannun et al., 2014 [4]), and bioinformatics (A. Esteva et al., 2017 [5]), to speech synthesis (A. van den Oord et al., 2016 [6]) and even game playing (V. Mnih et al., 2015 [7]), deep learning has become one of the most powerful tools. Convolutional Neural Networks (CNNs), which are biologically-inspired models (Y. LeCun et al., 1989 [8]) able of reaching great performance in image classification (A. Krizhevsky et al., 2012 [9]), natural language processing

(W. Yin et al., 2017 [10]) or analysis of time-series data, are a great example of successful models. There is a here a great opportunity to port the successes of deep learning in these various fields to benefit the world of Healthcare. With deep learning models, an increasing range of challenging tasks can be overcome, thanks to the possibility of using the knowledge from a larger pool of patients, like has been demonstrated for the detection of diabetic retinopathy in retinal fundus photographs (V. Gulshan et al., 2016 [11]). In sleep staging, SLEEPNET (S. Biswal et al., 2017 [12]) has been successfully deployed at Massachusetts General Hospital and helps in the real-time monitoring of sleep.

In this period of growth for the Health Informatics world, this thesis explores the use of deep learning models for the classification of seizures. Roughly 30% of critically ill patients in ICUs suffer seizures or related patterns of harmful electrical activity of the brain. Deep learning models could advance both the clinical value of brain monitoring and provide valuable scientific tools for studying seizures and related pathological cEEG events. However, there is an additional constraint to using deep learning models in healthcare applications, which is the need for interpretability. It would be hard and unacceptable to blindly trust the output of a black-box model when taking health decisions. Instead, informed decisions, supported by the output of an interpretable deep neural classifier, trained on large amounts of data, offers much better prospectives for the healthcare world. Interpretability is therefore a central objective throughout this study.

Together with the objective of obtaining accurate, interpretable classifiers, comes the need to design efficient methods for labeling a large volume of data, so that models can be properly trained. However, the classification of cEEG patterns is a challenging task, and even though concepts are well-defined and experts can confidently label clear-cut examples, some real-world sequences cannot be labeled with absolute certainty. With this additional constraint, obtaining a large dataset for training of deep neural networks is extremely challenging. Therefore, the objective of accurate classification of EEG patterns is interleaved with an even bigger focus on the design of an efficient method for label acquisition and

expert learning.

As a result, a co-learning technique is introduced, which allows to iteratively improve both dataset quality and understanding of the concepts involved in assigning a class label. This allows experts to become more consistent in the labeling task, while benefiting from interpretable feedback from the model.

CHAPTER 2

PERFORMANCE OF STANDARD LABELING

2.1 Introduction

Deep neural networks, and in particular deep Convolutional Neural Networks (CNNs), are generally successful at an increasing range of challenging tasks. In particular, this work focuses on the classification of seizures from continuous electroencephalograms (cEEG). Such recordings can be seen as multivariate time-series, on which CNNs give great performance (Y. Zheng et al., 2014 [13]). More specifically, published work shows that CNNs can be used for classification of EEG data (P. Bashivan et al., 2015 [14]). Therefore, they are a natural choice for approaching the classification of seizures.

Seizures are the result of abnormal electrical activity of the brain. They may result in convulsions in some cases, but most often go unnoticed. Epilepsy is the associated disease, in which seizures keep coming back. In 2015, an estimated 1.2% of the total U.S. population had active epilepsy, which represented at the time about 3 million adults and 470,000 children (M. Zack et al., 2015 [15]). This health concern comes with a great cost for the healthcare system, and both could benefit from automated monitoring of patients. In the Intensive Care Unit, most seizures go unnoticed. Since patients do not exhibit any kind of convulsions, diagnosis of seizures becomes difficult. Continuous monitoring is the only way to detect such seizures. Thus, this problem is a great candidate for the application of deep learning.

However, the first step involved in the training of deep neural networks consists in obtaining a high-quality labeled dataset, if no such dataset already exists. The traditional method for obtaining a dataset consists in human experts observing the input before assigning a class label and moving on to the next input. This simple and straightforward

technique, while being applicable to a wide range of domains, has a few limitations that make it unscalable and unreliable when the labeling task is challenging and expensive. The classification of seizures from EEG recordings belongs to such complicated tasks that cannot benefit from standard labeling techniques.

This first step in the present work analyzes an attempt to apply a standard labeling framework for our application of seizure classification, and outlines its numerous shortcomings. A more scalable, cluster-based labeling algorithm is evaluated as a way to overcome one of the reported issues. Conclusions from this preliminary work motivated further work on labeling methods in the second part of this thesis.

2.2 Dataset

In this section, the dataset structure is introduced along with statistics that illustrate the disagreement that exists among different graders.

2.2.1 Data Acquisition

Continuous EEG recordings were performed with an EEG cap with 19 electrodes. The location of electrodes on the EEG cap is shown on figure 2.1. On average, each recording lasts several hours, up to one or two days. These recordings have been provided by the Neurosciences Intensive Care Unit (ICU) at Massachusetts General Hospital (MGH).

2.2.2 Structure

From all the recordings, a total of 5103 sequences of 10 seconds each has been isolated into a dataset D_0 by selecting sequences that are representative of the ensemble of recordings, using a clustering technique. These 5103 sequences come from a total of 100 different patients.

Each sequence can be labeled in one of the eight following labels:

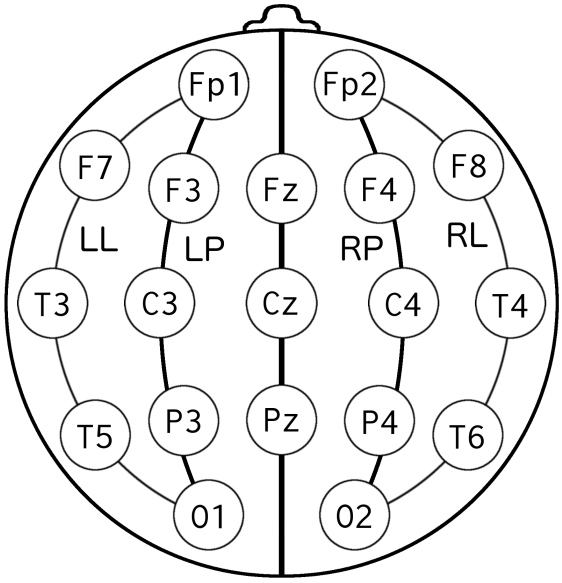

Figure 2.1: Location of electrodes on 19-channel EEG cap seen from above.

- Seizure (`SZ`)
- Lateralized Periodic Discharges (`LPD`)
- Generalized Periodic Discharges (`GPD`)
- Generalized Rhythmic Delta Activity (`GRDA`)
- Lateralized Rhythmic Delta Activity (`LRDA`)
- Burst Suppression (`BS`)
- Artifact (`AR`): defined to contain recordings that should be discarded, for instance at the very beginning of the recording or at the end
- Other (`O`): defined to contain normal brain activity

Table 2.1: Bipolar montages for each brain area.

LL	LP	RP	RL
Fp1−F7	Fp1−F3	Fp2−F4	Fp2−F8
F7−T3	F3−C3	F4−C4	F8−T4
T3−T5	C3−P3	C4−P4	T4−T6
T5−O1	P3−O1	P4−O2	T6−O2

2.2.3 Labeling Process

Each of the 5103 sequences in D_0 has been manually labeled by three experts. They were shown the signals computed from 16 bipolar montages, effectively translating the raw signal from 19 electrodes into a more meaningful representation, lasting 10 seconds each. The bipolar montages that we used are shown on table 2.1, the output being the difference between the two electrodes associated with a montage.

Additionally, the spectrogram for each brain area, lasting 10 minutes, was also provided during this labeling task.

2.3 Analysis of the Labels

2.3.1 Class Label Distribution

Here is presented the distribution of labels in the data. Table 2.2 shows the number of sequences labeled in each class by each of the three graders.

Apart from the main disagreement on *Other* and *Artifacts* most often labeled as *Other*, there is still a clear disagreement on clearly defined classes. Mostly, LPD is heavily confused with SZ.

Table 2.3 shows how many inputs each graders agree upon in terms of percentage. This means for how many sequences both graders assigned the exact same label out of the whole

Table 2.2: Class distribution of D_0 according to three graders.

Class Label	Score 1	Score 2	Score 3
LPD	439	531	655
LRDA	536	336	355
GPD	584	556	1002
GRDA	653	426	450
BS	186	222	375
SZ	1309	1340	65
O	1075	1692	2201
AR	321	0	0

Table 2.3: Percentage of agreement between each pair of graders.

Graders 1-2	Graders 1-3	Graders 2-3
45.01%	30.67%	49.23%

dataset. For the best combination of graders, this agreement falls just below half the input sequences.

2.3.2 Analysis of Disagreement

In order to better understand the disagreements in terms of labeling, the Cohen's kappa coefficients (J. Cohen, 1960 [16]) for each pair of grader is shown on table 2.4. In comparison with the percentage of agreement shown in table 2.3, the Cohen's kappa coefficient takes out the chance factor that could lead to an agreement where it is just the result of luck. As a result, the kappa coefficients are lower.

Additionally, an exhaustive representation, shown on figure 2.2, displays all the inputs that have been labeled differently by all three graders – 791 sequences, 15.5% of the dataset.

Table 2.4: Cohen's kappa coefficient for each pair of graders.

Graders 1-2	Graders 1-3	Graders 2-3
33.12%	18.54%	36.83%

On this experiment, *Artifacts* have been merged with *Other*. Each of the seven squares show the sequences labeled by the first grader who gave *Score 1*. The y-axis represents the number of sequences. On each plot is shown, for each sequence, how it was labeled differently by a second grader, assigning *Score 2* and a third assigning *Score 3*. Note that the thin colored line at the bottom of each plot does not show an agreement and should be ignored.

All these results clearly illustrate the complexity of classifying real-world EEG patterns and applying concepts to real data. Even if the concepts for each class are clearly defined, their application when it comes to classifying real-world EEG data is really challenging.

2.4 Alternatives to Manual Sequence Labeling

2.4.1 Shortcomings of the Standard Method

This first study confirms that the present way of labeling a dataset, when the task is challenging, is not scalable. This is for the following reasons:

- Obtaining the labels for this first dataset D_0 was heavily costly, experts spending too much time on each sample.
- Obtaining a dataset of sufficient size for the training of deep neural networks would require a lot of work and would incur large labeling expenses.
- Considering such budget is available, discrepancies in the labels and disagreements among human graders would substantially limit the performance of models trained

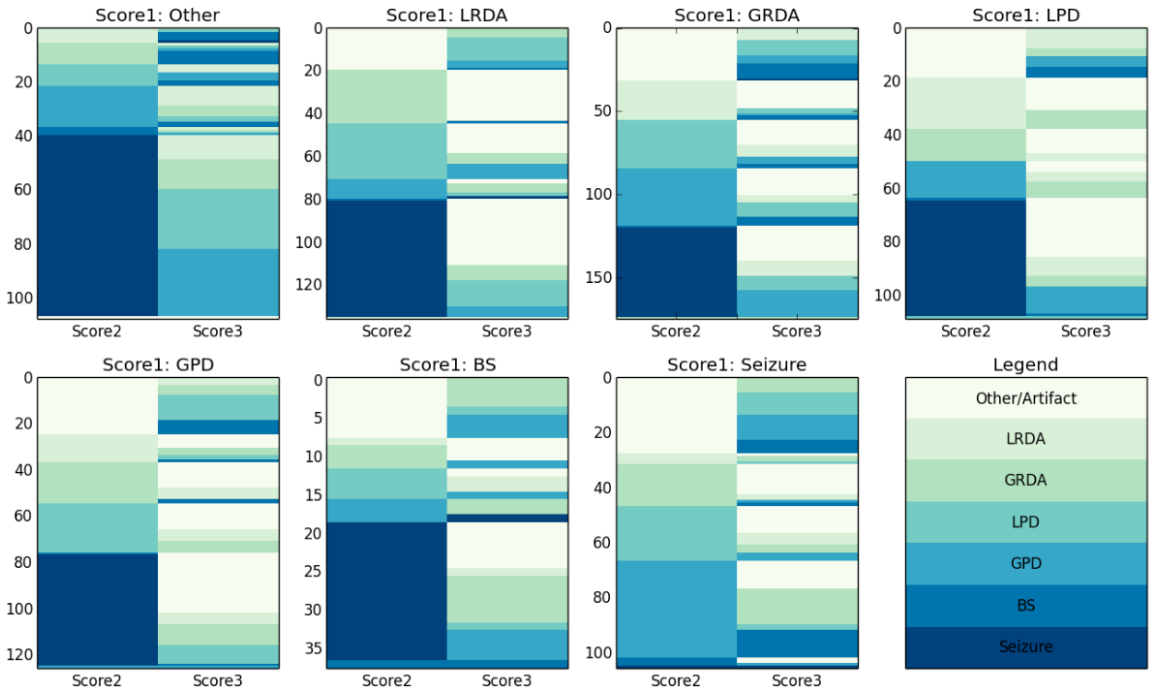

Figure 2.2: Detailed labeling disagreement in D_0

on the dataset.

2.4.2 Efficient Label Acquisition Technique

Knowing the issues that come with a standard labeling process, where experts look at each input separately, under restricted budget and for a challenging task, a method for scaling up the labeling efficiency was used instead of standard labeling. The idea is to cluster sequences by batches of similar input sequences. The method, described in algorithm 1, was employed at MGH. X_i represents the set of sequences $x_{i,j}$ for patient i.

Algorithm 1: Data Labeling

Input: Dataset $D_0 = \{X_1, \ldots, X_N\}$
Output: Labels $y_{i,j}$ for each $x_{i,j} \in X_i,\ \forall\, i \in [1; N]$

```
foreach i in [1; N] do
    Partition all x_{i,j} from X_i into k clusters C_{1...k}
    foreach c in [1; k] do
        repeat
            s ← x_{i,J} randomly taken from cluster C_c
            Observe sequence s
        until expert is confident about cluster label
        Choose class label L for this cluster
        foreach x_{i,j} in C_c do
            y_{i,j} ← L
        end
    end
end
```

The clustering used is based on a method similar to bag-of-words, that uses handcrafted features from the sequences. Similar sequences, based on this clustering, are grouped together and assigned the same label.

The method quickly generates labels for all the available data at a very low cost. However, a large part of these labels now need to be refined, since the clustering leads to mistakes in the process. However, this method is sufficient to train a first classifier, which can in turn be used to help the experts re-evaluate some sequences.

2.5 Conclusion

With this first work, it is clear that obtaining a labeled dataset for the classification of brain activity into several types of seizures is both a costly and challenging task. Furthermore, traditional techniques for obtaining a dataset do not apply here: these standard techniques are too costly, do not scale properly and poorly fit the difficulty of the task.

A different approach can be taken. Scale is the first shortcoming of traditional techniques that can be easily overcome. A clustering-based algorithm is used for this purpose. However, the labels generated remain low-quality labels that need to be refined.

In the following chapter, a general technique for efficient label refinement is introduced, that allows to obtain a labeled dataset of great quality at reduced cost, for the ultimate training of deep neural classifiers. In addition, the method allows a better understanding of the concepts over time, leading to less classification mistakes on the most challenging inputs, that cannot be achieved with a traditional method.

CHAPTER 3

INCREASING LABELING EFFICIENCY: A CO-LEARNING TECHNIQUE

3.1 Introduction

Deep learning has become increasingly successful at even the most challenging tasks, from image classification (K. He et al., 2015 [3]) to game playing (V. Mnih et al., 2015 [7]). However in supervised learning, excellent performance always comes with the same burden, regardless of the field of research: the need of large quantities of high-quality labeled data. The problem of detecting events in electroencephalograms (EEG) belongs to this class of applications where deep learning can achieve excellent results but requires large amounts of labeled data.

However, in many real world applications, the labels are difficult to acquire. Either the acquisition costs are too high to make it possible to collect enough data, or events of a certain label are simply too rare to be observed enough times. In both situations, it becomes hard to apply deep learning algorithms. In other recurring situations, there is an abundance of raw data, but a lack of high quality labeled data, again due to either high labeling costs or the difficulty of the labeling task. In biomedical applications, data acquisition is a first challenge to be overcome.

First, for privacy reasons, it can be difficult to obtain patient data. Second, labeling is often expensive in that it requires availability of a domain expert who can dedicate enough time to the dataset creation. Third, tasks can become challenging to the point where even domain experts cannot readily come to an agreement on some or many sample points (N. Gaspard et al., 2014 [17]). For a classification task, they might often disagree on difficult sample points at the boundaries of multiple classes, or have different interpretations of the established concepts. For these reasons, obtaining a large dataset with high-quality labels

is quite challenging.

As an example, one great challenge in biomedicine is related to the classification of seizures, which are the result of abnormal electrical activity in the brain. There exist multiple types of seizures, and classifying them allows to study their respective impact on health, as well as how to effectively cure or treat them. A recent study by Ruiz et al. [18] showed that the analysis of continuous electroencephalography (cEEG) signals can help predict the risk of seizures in critically ill patients. cEEG is a non-invasive method to monitor the electrical activity of the brain. In the critical care setting, cEEG monitoring is typically performed for 24-72 hours at a time, providing large volumes of data. However, manually labeling the events is a tedious task for the human expert, both due to the high volumes of data and the difficulty of the task.

Deep learning models could advance both the clinical value of brain monitoring and provide valuable scientific tools for studying seizures and related pathological cEEG events. Deep neural networks have already been used to study EEG patterns (P. W. Mirowski et al., 2008 [19]; P. Bashivan et al., 2015 [14]), so they are a model of choice in our study. The issue here is the costs related to labeling data, making it hard to rely on a standard learning framework. In addition, EEGs are difficult for experts to label consistently due to the frequent overlap between classes (JJ. Halford et al., 2015 [20]; H. A. Haider et al., 2016 [21]). The standard solution to this problem in the medical literature is to have each sample reviewed and labeled independently – or in a committee – by multiple human experts before a decision on the class label is taken. This approach is however not scalable. Overall, this situation makes it challenging to obtain a labeled dataset suitable for proper training of deep neural networks, and therefore is a great application of our work on label acquisition.

With the help of active learning, it has been shown that a classifier can be trained on a judiciously selected subset of the available data while performing as accurately as models trained with a much larger set of randomly selected training examples (K. Wang et al., 2017 [22]). Such efficient methods for selecting sample points for human labeling allow

to cope with budget restrictions with no or limited trade-off on performance. However, although a similar framework could help us efficiently grow our dataset, all active learning methods start with the assumption that they can query an "Oracle", a human expert who can label any sample point – the query – with no risk of misclassification. In many studies, the "Oracle" is a human, expert in the field, and the correct class label leaves no room for doubt. In other studies, like ours, the problem is so challenging that fully relying on human experts in order to obtain ground truth class labels is not enough. Indeed, the task is so difficult that there is a great risk that the label given by the human expert may still be wrong. Therefore, although active learning remains attractive for our study, using it would not solve our issues.

Instead, we first acknowledge that in an increasing number of studies, although the model can be seen as being taught by humans, it is not uncommon that the overall accuracy of the model is better than that of all human teachers combined (V. Gulshan et al., 2016 [11]; D. Silver et al., 2017 [23]). In this work, we use this fact as a way to improve the human raters' performance and consistency in the labeling task, especially when labeling data that spans multiple different patients, and we propose `HAMLET`, a Human And Machine co-LEarning Technique to efficiently bypass this label acquisition difficulty. We apply `HAMLET` to train a classifier for various types of non-convulsive seizures, based on continuous EEG recordings. `HAMLET` helps us face the above mentioned issues, while making use of the great performance of deep learning models at challenging tasks to improve label quality in a feedback-loop fashion. `HAMLET` fully integrates the limiting constraint of lacking an "Oracle" or absolute source of truth. With `HAMLET`, we were able to improve our dataset, ultimately obtaining higher classification accuracies. Our contributions can be summarized as follow:

- Design of an algorithm for efficient label improvements.
- Increased interpretability of our classifier with the use of a separate memory module hosting representative reference embeddings.

- Successful application of our method to the challenging task of seizure classification.

HAMLET has shown significant performance gain against deep learning and other baselines, increasing accuracy from 7.03% to 68.75% on challenging inputs. Besides improved performance, clinical experts confirmed the interpretability of those reference embeddings in helping explaining the classification results by HAMLET.

In this article, we first survey related work in EEG classification, deep learning for health informatics, active learning and co-training. Our co-learning technique and the architecture of our classifier are introduced in section 3.3. Finally, in section 3.4, we present our dataset, methods for pre-processing, and results.

3.2 Related Work

3.2.1 EEG Classification

Using Feature Extraction

The topic of EEG classification is a fertile area of research. Most methods traditionally rely on feature selection combined with a classifier, such as Support Vector Machines (SVM), Linear Discriminant Analysis (LDA) or k-Nearest Neighbors (k-NN) (M. H. Alomari et al., 2013 [24]; H. Shoeb et al., 2010 [25]). Relevant features include Common Spatial Patterns (CSP), Filter-Bank Common Spatial Patterns (FBCSP), and Logarithmic Band Power (BP) (X. Yong et al., 2015 [26]). Seizure classification has also been approached with feature engineering (F. Fürbass et al., 2015 [27]) for instance using signal amplitude variation (AV) and a regularity statistic (PMRS) (J. C. Sackellares et al., 2011 [28]). Although careful feature engineering can lead to great performance, it is not be strictly necessary.

Deep Learning

Classification of EEG data can be seen as a multivariate time-series classification problem (P. Bashivan et al., 2015 [14]). Furthermore, one advantage of CNNs is the automated

feature selection that happens during the training process. Without additional work, the model learns the features that it finds most relevant for its given task, from the raw signals given as input. This has been shown with great success for sleep staging (S. Biswal et al., 2017 [12]) as well as seizure classification (U. Rajendra Acharya et al., 2017 [29]). However, all these deep models require high volumes of labeled data for training, which is the bottleneck in our application. Therefore, CNNs remain a model of choice but we cannot use them directly in the present study due to the label limitation.

3.2.2 Convolutional Auto-Encoders (CAEs)

CNNs can be used in a supervised learning framework, but cannot benefit from the large amounts of unlabeled data. Traditionally, auto-encoders (AEs) have been widely used to extract meaningful features in the absence of labels even on biomedical data (J. Tan et al., 2015 [30]). However, in their simplest form, AEs cannot be efficiently applied to time-series as they ignore their bi-dimensional structure. Convolutional Auto-Encoders (CAEs) bring together the advantages of using convolutions and auto-encoders (X. Mao et al., 2016 [31]; Masci et al. [32]). However, seizures are typically relative rare events, whereas CAEs tend to learn to represent the dominant statistical structure of the underlying data, which is not directly relevant for seizure classification. Therefore using CAEs alone is not sufficient in our classification task.

3.2.3 Co-Training

Co-training is a semi-supervised machine learning algorithm, where two models can be concurrently trained, each with its own set of features. Additionally, the predictions of one model on an unlabeled dataset are used to enlarge the dataset of the other model. The algorithm has originally been introduced for the classification of web pages (A. Blum et al., 1998 [33]). Here we do not focus on increasing dataset size but rather label quality within the labeled dataset. Although the standard setting for co-training includes unlabeled

data that models can benefit from, it has been shown that co-training can still save labeled samples even in a strictly supervised setting (M. Darnstädt et al., 2009 [34]) which we use. In the original co-training algorithm, there is the possibility that the training set of a model is augmented with wrong labels from the other model. COTRADE improves on this idea by only allowing models to share labels they are confident about (M. L. Zhang et al., 2011 [35]). This notion of labeling confidence is at the center of `HAMLET`, however, there are some differences in our approach to co-training. Indeed, one of the two entities at the center of the algorithm is the human expert. The expert knowledge helps training the algorithm by updating the labeled dataset, while the model feedback helps improving the consistency of the expert at the classification task. Because of the human learning dimension, we use the term co-learning instead.

3.2.4 Active Learning

In situations where unlabeled data is abundant, but labeled data is scarce due to a restricted labeling budget, active learning offers a way to select the most informative sample points in order optimize labeling resources on the data that is most helpful for the model. Heuristics for active learning include Query by Committee (QBC), uncertainty sampling, margin sampling, entropy or Expected Gradient Length (EGL) (B. Settles, 2010 [36]). Although these methods might reduce the need for large label sets, this still is not enough for deep learning models like CNNs. For such models, including pseudo-labeled high-confidence data into the training set is a possible approach (K. Wang et al., 2017 [22]). Unfortunately, active learning implies the existence of a source of truth that is not available in our challenging classification task, since in our study, even humans find it difficult to label many of the patterns commonly encountered in EEGs of critically ill patients.

3.2.5 Using Memory Modules in Neural Networks

Recently, researchers have been experimenting with augmenting neural networks with memory modules. For instance, a memory module is present in the architecture of Matching Networks (O. Vinyals et al., 2016 [37]). In their model, the module is used together with an attention mechanism, however without interpretability in mind. Our design, presented in the following section, emphasizes interpretability thanks to an external memory module.

3.3 HAMLET Method

3.3.1 Background

Before detailing our implementation, we give a brief description of common models used in this work.

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (Y. LeCun et al., 1989 [8]) are one of the most common model for image classification, but find applications in many other fields. A CNN is made of a succession of layers that allow automatic feature learning by the model. Although a lot of variations are available in order to tweak the model for specific needs, the basic structure of each layer is a succession of a convolution layer followed by a pooling operation.

A convolution layer applies a convolution to its input X to generate K pre-activation maps $H_{1...K}$, learning K kernels W_i and K biases b_i along the way:

$$H_i = W_i * X + b_i, \ \forall \ i \in [1; K]$$

The shape of a kernel depends on the input data, available computing resources and depth – number of channels – of the input feature matrix X. Essentially, each convolution layer

has only a limited number of parameters, since each kernel is shared across all the input surface. These pre-activation maps go through an activation function, typically a Rectified Linear Unit (ReLU) (V. Nair et al., 2010 [38]), allowing the model to learn non-linearities.

Pooling layers effectively reduce the dimensionality of the data with simple deterministic operations. The output of a pooling layer is a smaller matrix where each sub-matrix of a specified size of the input, is replaced by, most often, the maximum value of this sub-matrix (max-pooling), or the average (average-pooling). Dimensionality can also be reduced by selecting an appropriate stride in the both layers.

After a succession of this basic building block of CNN, the dimensionality is greatly reduced, and the network has learned important features of the input, independently of their location thanks to the use of convolutions, and with few parameters.

Finally, the output of the last layer can be flattened into a vector and fed to a MultiLayer Perceptron (MLP), for the final classification task. An MLP is also made of a succession of layers, though usually only very few hidden layers are used. Each layer of the MLP consists in a weight matrix W and a bias vector b. Given input vector x, each layer outputs y:

$$y = \sigma(Wa + b),$$

where σ is an activation function. When the model is used for a classification task, the last layer is made to output y of dimension the number of classes of the problem. The model can be trained to output 1 at the corresponding class label and 0 everywhere else – a technique known as one-hot encoding –, and the decision of the model is the index of the neuron with highest activation value.

Convolutional Auto-Encoders (CAEs)

Convolutional Auto-Encoders are built with the same building blocks as CNNs, however their structure is symmetric. They are made of two parts: an *encoder* and a *decoder*. The

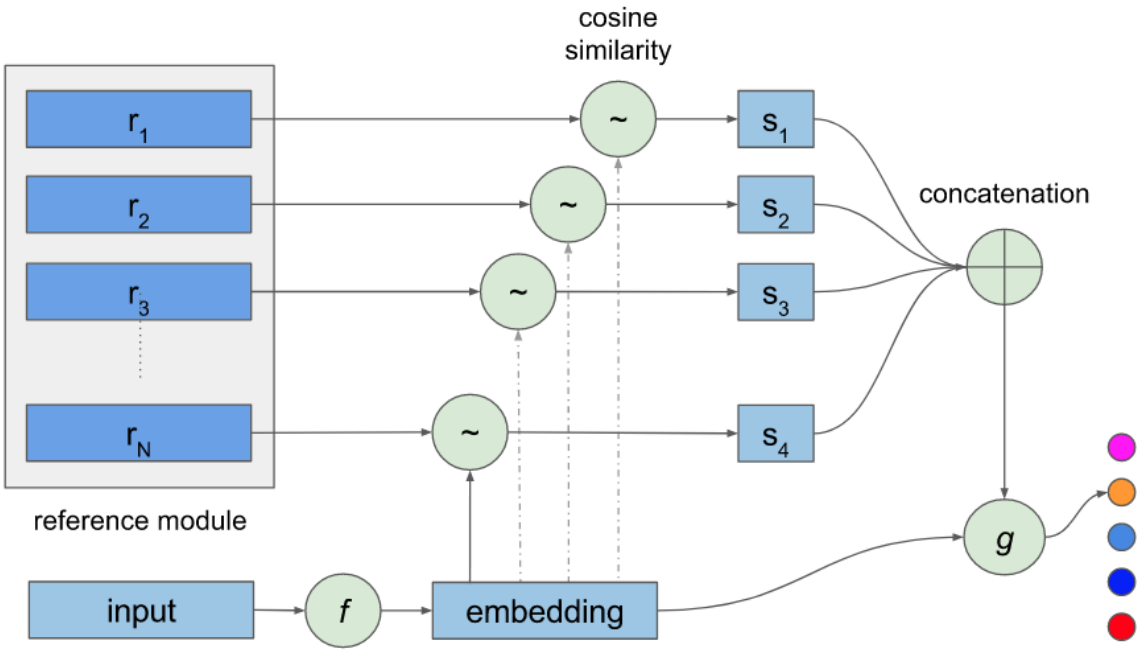

Figure 3.1: Model structure with embedding function and memory module.

encoder learns the features that are the main characteristics of the input. The output of the encoder is a low-dimensionality representation of the input, i.e. the embedding. The decoder that follows is assigned the task of reconstructing the original input as closely as possible. It can either learn its own kernels or apply the opposite kernels of the encoders. The loss is a measure of the difference between the original input and reconstructed output.

3.3.2 Co-learning Framework

Architecture of the Classifier

One important part of our HAMLET framework is the classifier, that is used to both improve the human expert through re-labeling suggestions, and of course the classification task itself, the ultimate goal of the whole process. Its architecture is shown in figure 3.1. It is based on a combination of the following three components:

- Embedding function f, either a CNN (supervised training) or a CAE (unsupervised)

- Memory module (pictured above the input on the figure) containing N reference embeddings used to compute similarity scores s_i
- Dense layer g, used as a final classification layer

One key novelty of our model lies in what we call a memory module. In this separate module, a set of N reference embeddings is stored. For a given input i, the encoder creates an embedding $e_i = f(i)$ that is compared to each reference embedding r_j using cosine similarity, giving N similarity scores s_j. These similarity scores and the current embedding e_i are concatenated to form the immediate representation fed to the classifier function g, typically a multilayer perceptron (MLP). We will explain in further sections how using reference embeddings enhances interpretability.

In our experiments, we used $N = 512$, and a dense layer g with one hidden layer of $n = 1024$ neurons. We propose two different embedding functions f: 1) a supervised alternative that is trained with the available labeled dataset; and 2) a Convolutional Auto-Encoder (CAE), which allows to benefit from the whole dataset.

Algorithm

Our co-learning framework is best described with the following algorithm. We start with a first dataset with low quality labels, and iterate through the following steps:

1. Fine-tuning (pre-training for the first iteration) of the embedding function f.
2. Selection of reference embeddings.
3. Learning of the dense layer g.
4. Label improvement through machine feedback.

The procedure is illustrated on figure 3.2. The model is first trained on the original training set TR_1 and evaluated on the testing set TE_1, keeping track of training and testing scores Sc_1. The expert evaluates the results, and updates a subset of both TR_1 and TE_1. A more

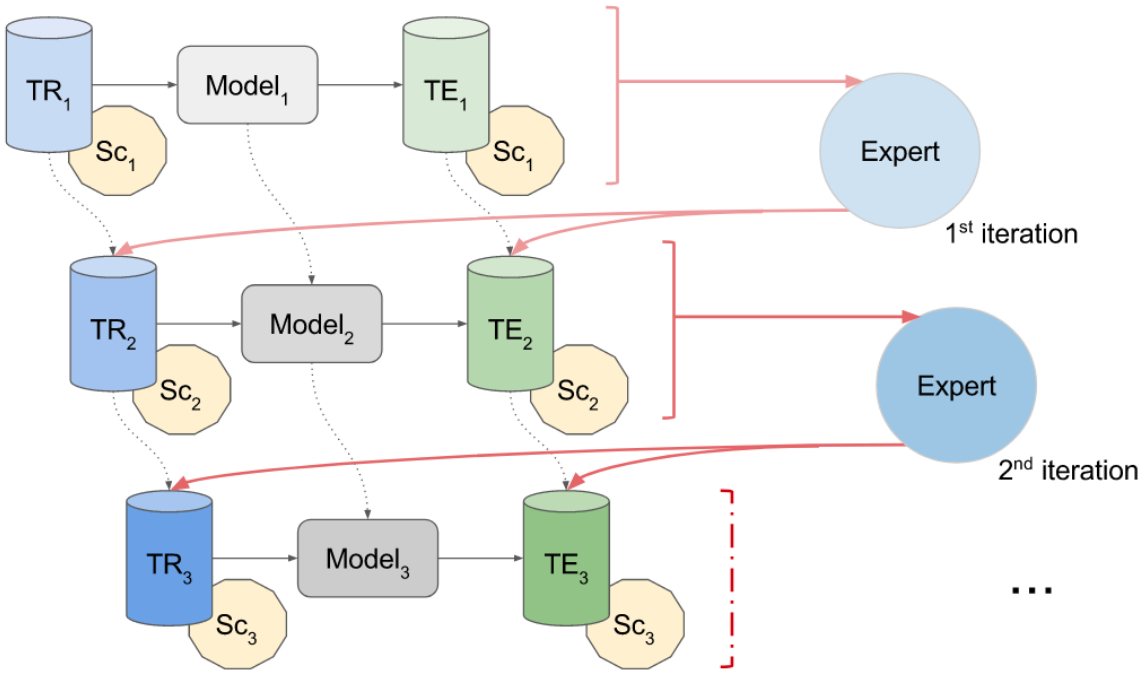

Figure 3.2: Co-Learning algorithm with machine feedback.

robust model is obtained by fine-tuning the original one using the newer version of the dataset. By going through multiple runs of the above algorithm, the model is iteratively improved thanks to the increase in label quality. Concurrently, the expert learns from past classification mistakes and becomes more consistent at the labeling task. Each phase of the algorithm is described in the following sections.

3.3.3 Fine-Tuning (or Pre-Training) of the Embedding Function

In this step, either a supervised or unsupervised approach can be taken. During the first iteration, the function f is trained from scratch on the dataset. Further iterations of this algorithm only fine-tune f and no pre-training is required.

Supervised Embedding Function

For the supervised approach, we experimented with a CNN. In this case, f corresponds to the first layers of the model up until the start of the dense layer. After training is complete, the dense layer is dropped.

The architecture of our CNN is shown at the top of figure 3.3. The first layer uses a depth-wise separate convolution in order to better handle the high dimensionality of input data in EEG studies – 16 montages – a practice inspired by recent work on EEG classification (R.T. Schirrmeister et al., 2017 [39]). The depth-wise separate convolution first performs a convolution through time (per channel), followed by a convolution across all electrodes (also known as `1x1` convolution). The following blocks are standard groups of convolution, max-pooling, and dropout layers. We have used a dropout rate of $p = 0.2$ and batch normalization, which has been shown to improve generalization (S. Ioffe et al., 2015 [40]) right after the convolution layers during training. We use Exponential Linear Units (ELU) which provide faster learning (D-A Clevert et al., 2015 [41]) as our activation functions. The dense layer has 1024 neurons in the hidden layer, and ends with 5 softmax units corresponding to our class labels. We also use this CNN as a baseline for classification accuracy.

Unsupervised Embedding Function

In the unsupervised approach, a Convolutional Auto-Encoder (CAE) is a great choice for f. A CAE is made of an encoder, which creates an embedding, and a decoder that takes this embedding and reconstructs the original sequence. After the CAE is trained on the whole dataset, the decoder is dropped, and f is the encoder.

Our CAE architecture is shown at the bottom of figure 3.3. Each input channel is 1D and parameters are indicated as {`width, stride`} on the figure. On the top is shown the supervised embedding function. Although not represented, batch-normalization is applied after convolutions and dropout after max-pooling layers with a rate of $p = 0.2$. The dense

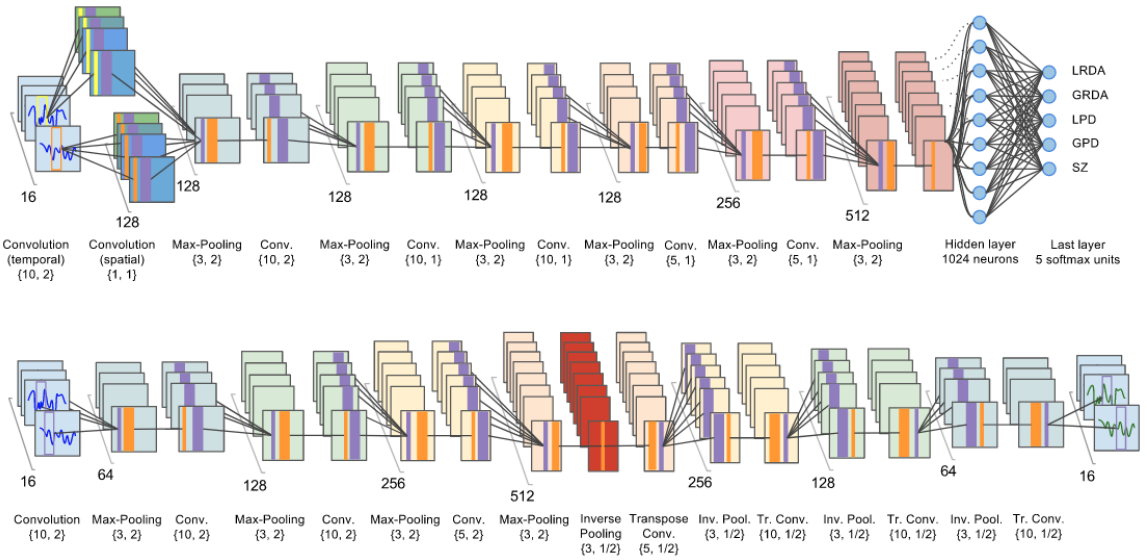

Figure 3.3: Embedding function f, supervised and unsupervised alternatives.

layer is only used for training, then dropped when used in HAMLET. On the bottom of the figure is shown the unsupervised embedding function. The first layers (before red boxes) are the encoding part of the network. They build the embedding (shown in red). The following layers are part of the decoder, which is only used for training, then dropped when used in HAMLET.

In the CAE, there is no dense layer. Instead all the operations are reversed in the decoder after the last max-pooling layer. To build the decoder, we use un-pooling layers which increase the sequence size from encoded representation to input length, and transposed convolutions (sometimes called de-convolutions) in order to keep the overall structure completely symmetric.

3.3.4 Selection of the Reference Embeddings

The reference embeddings are selected right after the embedding function f is learned. It is crucial that they are a close representation of the input data, and diverse enough to allow the model to use the module for all the classes present in the dataset. To ensure

diversity, we use k-means among inputs of a same class, giving equal space in the memory module to all classes. Compared with a k-means on all available inputs, some reference embeddings might not be as useful for the model, or might even be redundant for rare classes, but this ensures that embeddings are always available for the model to learn from. For interpretability purposes, we keep track of the labels and sequences that correspond to the selected embeddings.

3.3.5 Learning of the Dense Layer

The next step is to teach the model to correctly classify sequences of EEG, using both the embedding function f and the memory module. The outputs of both components are concatenated and fed to the classifier g. The model is then trained on the labeled dataset in a supervised manner.

3.3.6 Label Improvement with Machine Feedback

The final step of training is where machine feedback comes into play. Using one of the machine feedback strategies described in 3.3.7, we suggest new labels for some sequences that the model mis-classified with high certainty. These suggestions are likely to propose sequences that have been mis-classified by the human grader. They are given to an expert for re-evaluation.

After a portion of the dataset has been re-labeled, the model is fine-tuned and new tests can evaluate performance increases. The labeling effort should be shared between samples from the training and testing sets. Although updating the testing set does not improve the model, it improves the quality of the experiment as a whole. A reasonable choice is to share this effort proportionally to each dataset size. However, in our experiments we also include results showing improvements on re-evaluated testing sequences only, to better display the improvements brought to the labels.

3.3.7 Machine Feedback Strategies

In traditional active learning, various heuristics can be used for suggesting new data to be labeled, with the intent of increasing the labeled dataset size. This is called uncertainty sampling. These heuristics include Least Confidence (D.D. Lewis et al., 1994 [42]), Margin Sampling (T. Scheffer et al., 2001 [43]) and entropy (C. E. Shannon, 1948 [44]). Each heuristic outputs samples that would be most informative for the model. Here, we are looking for misclassification points that our model is most confident about, in order to suggest potential label errors by humans. Therefore we modify the active learning heuristics for improving label quality during the co-learning process. Below is a description of the various strategies for co-learning.

Highest Confidence

Using the confidence of the model accuracy on each piece of input data, we can select inputs that the model is most confident about. For each input i, the confidence c_i is directly given by the probability of the class with highest probability according to the model: $c_i = \max_j p(y_i = j)$ for each class j. After all confidence values c_i are obtained, they are ranked in decreasing order. For inputs i with high confidence values c_i that were wrongly classified by the model, a mistake on the original label is very likely. Therefore, those points will be provided as machine feedback to human experts for relabeling.

Margin Sampling

The margin sampling heuristic is also based on confidence values. Inputs with the highest difference between the confidence values of their two most likely classes indicate high confidence. For input i: $m_i = p(y_i = j_1) - p(y_i = j_2)$, where j_1 and j_2 are the two most likely labels, according to the model. The misclassification points with large margin will be used as machine feedback.

Entropy

Finally, entropy can also be used as a measure of the certainty for each input sequence. The entropy of all sequences are sorted in increasing order, the lowest values being the least informative ones – i.e. those the model is most confident about. For a given input sequence i, the entropy e_i is given by:

$$e_i = -\sum_{j=1}^{C} p(y_i = j) \log(p(y_i = j)$$

3.3.8 Interpretability of the Model

When using the model to perform a classification task on new data, the memory module can be used to explain the reasoning of the model. For a given input i, N similarity scores s_k will be generated from the memory module, one per reference embedding r_k.

In our experiments, we show that `HAMLET-CNN` learns to effectively use the reference embeddings for classification. However, we expect `HAMLET-CAE` to lack interpretability, as unsupervised training only teaches the model to recognize features that are only relevant to sequence encoding. As a result, `HAMLET` uses the memory module as a bank of features more than as a way to discriminate between classes. Therefore, interpretability can only be claimed for the supervised `HAMLET-CNN`.

Model interpretability allows to better justify the label suggestions in the last phase of the algorithm. For a given input sequence, if we look at the closest embeddings within the memory module – those with highest similarity scores – we will most likely reach a reference sequence that shows a similar pattern to the current one. Formally, to justify the decision of the model to classify i into class c, we can look at the reference embedding $r*$ with highest similarity score among reference embeddings for the same class c:

$$r* = \arg\max_{r_k \in c} s_k, \quad s_k = \text{cosine\_similarity}(r_k, f(i)).$$

At this point, we can show the labels previously assigned to that reference sequence – preferably by the same expert – and explain why the model made such a suggestion. This increased interpretability enhances co-learning in the two following ways:

- First, the expert knows the model did not randomly happen to output a given class label. This decision is supported by interpretability and is far less likely to be ignored by the expert.
- Second, by reminding experts how they previously labeled similar sequences, they can learn much faster and become more consistent while labeling.

3.4 Experiments

3.4.1 Dataset

Acquisition of Continuous EEG Recordings

EEG recordings are multivariate time-series describing the electrical activity of the brain. In this study, they are recorded in a non-invasive manner by affixing 19 small metallic (silver/silver chloride) electrodes directly to the scalp. The EEG electrodes were placed in standardized locations, following the international 10-20 system, with locations and labels of electrodes as shown in figure 2.1. On the figure are represented the different brain areas. `Fz`, `Cz` and `Pz` are reference electrodes. The bipolar montages that we used are shown on table 2.1, page 7 in the previous chapter. There are four montages for each of the brain areas: Left Lateral (LL), Left Posterior (LP), Right Posterior (RP), Right Lateral (RL). Each montage is the difference between the channels.

Our original dataset, provided by the Neurosciences ICU at Massachusetts General Hospital (MGH), contains multiple hour-long (generally over 24 hours) recordings of EEG for 155 different patients, sampled at $f = 200$ Hz. There is a plan to release the de-identified dataset to the public in the near future. In this dataset, we denote the EEG of

patient i by $X_i \in \mathbb{R}^{d_e \times m_i}$, with $d_e = 19$ the number of electrodes and m_i the length of the given EEG.

Pre-processing

After acquisition, the raw EEG data goes through the following pre-processing pipeline:

- Low-pass filtering: the raw signals are first filtered with a 60.0 Hz low-pass filter, so that high-frequency noise and electrical artifacts are reduced.

- Computation of montages: it is usual to work with a bipolar montages instead of the raw data from the electrodes, to reduce interference from electrocardiogram (EKG) signals. This is done according to the table in figure 2.1, page 6 in the previous chapter. Let $M_i \in \mathbb{R}^{d_m \times m_i}$ be the montages for X_i for patient i, with $d_m = 16$ the number of montages generated in the process – i.e. channels.

- Splitting of recordings: all EEG recordings are split into 16-second sequences. We denote each sequence from X_i and M_i by $s_{i,j}$ and $m_{i,j}$, respectively. Because the patterns representative of each class are usually clear only within a larger contextual time window, experts look at an additional 6 seconds of signal on each side of the sequence when labeling.

Initial Labeling Process

We have obtained the initial labeled recordings of 155 patients, for a total of 4176 hours, or an average of 27 hours per patient. Each sequence from these recordings has been manually labeled by a clinical expert. As we described about this task, high error rate is expected in these initial labels.

Each sequence of EEG can be given one of six class labels, where five correspond to the patterns of brain activity of primary interest (Seizure, Lateralized Periodic Discharges (LPD), Generalized Periodic Discharges (GPD), Generalized Rhythmic Delta Ac-

Table 3.1: Number of 16-second sequences from each class in the labeled dataset D.

Class Label	Sequence Count	Percentage
Seizure	128,691	33%
LPD	115,729	30%
GPD	84,168	22%
GRDA	32,566	8%
LRDA	29,332	7%
Total	390,486	100%

tivity (GRDA), Lateralized Rhythmic Delta Activity (LRDA)), and the last one corresponds to Other/Artifacts (O/A). A great majority of the recordings is made of either background activity, noise, and non-physiological artifacts (O/A), so we put such sequences aside. In real-life, we can easily automate this selection step with a binary classifier. Let D be the labeled dataset we obtain at this stage, without O/A (D contains 5 classes).

Creation of Balanced Datasets

For our experiments, we created three datasets from all the available 16-second sequences $s_{i,j}$ in D. The class distribution in the full dataset D is highly skewed, as shown in table 3.1, so we keep the datasets balanced in terms of class labels to ensure the models do not learn trivial frequency bias:

- D_{20k}^{unseen}: 20,000 sequences (89 hours of EEG data), split into a training set (80%) and a testing set (20%). Patients in the testing set are not present in the training set (testing is performed on *unseen* patients).

- D_{20k}^{known}: 20,000 sequences (89 hours of EEG data), split into a training set (80%) and a testing set (20%). Patients in the testing set are also present in the training set (testing is performed on *known* patients).

- D_{100k}: made of 100,000 sequences from D that are not in the previous two datasets. This larger dataset represents 445 hours of recordings and is only used for unsupervised training of the embedding function f (CAE). Patients in the testing set are not present in the training set.

Dataset Augmentation

In this classification task, the most challenging issue is to make the model learn how to generalize across new patients. During training, in order to simulate different patients, the electrodes from the left and right side of the brain can be flipped, while the three reference electrodes in the middle of the scalp – `Fz`, `Cz` and `Pz` – remain unchanged. Our classification task being a symmetric problem, which particular side of the brain exhibits a pattern does not affect classification. This simple technique almost duplicates the training dataset in terms of the number of patients.

3.4.2 Setup

The model has been trained using a server with Intel(R) Xeon(R) CPUs E5-2630 v3 running at 2.40 GHz, 32 cores, with 256 Gb of RAM and 4 GPUs Tesla K80, NVIDIA Corporation GK210GL, with CUDA v8.0. Training has been performed with Python version 2.7, using version 1.4.1 of `Tensorflow`. Other libraries needed for `HAMLET` include `numpy` and `scipy`. With the above configuration, training `HAMLET-CNN` during 100 epochs on D_{20k}^{unseen}, with a batch size of 128, took thirteen hours. We used the `Tensorflow` implementation of the stochastic optimizer *Adam* (D.P. Kingma et al., 2014 [45])

3.4.3 Evaluation

Next, we evaluate our classification models and `HAMLET` technique. For each experiment, we have trained our models for 100 epochs, saving the model with highest accuracy on the testing set.

Classification

To get a sense of the difficulty of the task, the performance of our models and various baselines has been evaluated in the following two scenarios:

- Known patients: with D_{20k}^{known}, where patients in the training are also present in the testing set, we first assess the ability of the model to classify known patients, which can be useful in some clinical situations such as long-term monitoring at ICU. In this setting, clinicians wish to capture seizure patterns similar to those observed previously, or in long-ambulatory long-term monitoring settings with implantable electrodes where there is the opportunity to fine-tune the system based on previous seizures.

- Unseen patients: using D_{20k}^{unseen}, we evaluate the performance of the model at generalizing across patients. It is important to ensure the model can classify EEG from different brains. This is an obviously harder task, leading to an understandable drop in accuracy when evaluating in this setting.

For each scenario, we have experimented with two variations of `HAMLET` with each $N = 512$ reference embeddings, either with supervised embedding (`HAMLET-CNN`) or unsupervised embedding (`HAMLET-CAE`), as well as the following baseline models:

- Convolutional Neural Network (`CNN`): this baseline is the model that we use as a supervised embedding function f, introduced in 3.3.3. Therefore, the complexity of this baseline is comparable to that of `HAMLET-CNN`.

- MultiLayer Perceptron (`MLP`): we have trained an MLP with 1024 neurons in the hidden layer to perform the same classification task.

The results of this experiment on both known patients and unseen patients, shown in table 3.2, confirm the existence of inconsistencies within the dataset. Although classifying

Table 3.2: Accuracy on the testing sets of D_{20k}^{known} and D_{20k}^{unseen}.

Model	Unseen Patients	Known Patients
HAMLET-CNN	39.36%	75.91%
HAMLET-CAE	38.46%	75.07%
CNN	38.89%	74.93%
MLP	21.04%	20.05%

Table 3.3: Accuracy on the testing set of D_{20k}^{unseen}, before and after re-evaluation.

Model	Before re-labeling		After re-labeling	
	full test	re-eval only	full test	re-eval only
HAMLET-CNN	39.36%	7.03%	40.75%	68.75%
HAMLET-CAE	38.46%	10.94%	39.06%	67.97%
CNN	38.89%	6.25%	41.58%	68.75%
MLP	21.04%	0.78%	23.14%	14.06%

EEGs of new patients is significantly more challenging, when labeling the sequences, experts usually remain consistent for sequences coming from the same patient. However, it is hard for them to remain consistent when labeling sequences from new patients having their own specific patterns. This is where re-labeling with machine feedback will show how HAMLET can be used to drastically improve performance.

Co-Learning

Results on the original D_{20k}^{unseen} dataset (before label re-evaluation) from the previous experiment are again shown on the first two columns of table 3.3. These accuracies set the starting point for improvements using co-learning.

In order to evaluate our co-learning framework, we ran one iteration of our algorithm,

providing a suggestion of sample points to be re-labeled by a human expert, using the highest confidence heuristic introduced in section 3.3.7. Results show that after this first iteration alone, during which 837 sequences (4.18% of the dataset D_{20k}^{unseen}) have been re-evaluated by an expert, accuracy on the testing shows great improvement. The results are shown on table 3.3, both for the full testing set, as well as on the subset of sequences re-evaluated during co-learning. We note a clear increase in model performance after this first iteration alone. In most scenarios, `HAMLET-CNN` performs better than the unsupervised embedding alternative `HAMLET-CAE`, and also better than `MLP` which cannot learn properly, as expected. Our `CNN` has good performance overall, but cannot claim to be interpretable like `HAMLET-CNN`.

The confusion matrices for `HAMLET-CNN` in the first iteration (trained and tested on the original testing set, at the top) and after re-labeling (after fine-tuning, with training and testing on the improved datasets, at the bottom) are shown together on figure 3.4. Each row shows, for a given expert-provided class label, how it has been classified and in what proportions by the model (a diagonal matrix is ideal). It is interesting to notice how the model improves based on these confusion matrices: although after co-learning, seizures are not as clearly recognized by the model as before, the model now better classifies LRDA and LPD, leading to overall better accuracy and classification performance of the model.

Interpretability

Finally, we analyzed how the model uses the reference embeddings when making decisions. This can be done by looking at the weights of the dense layer that are applied to the reference embeddings. For each output class label c, we selected the 16 reference embeddings with biggest weights in the dense layer, among the 512 available embeddings. We then computed what percentage of these 16 embeddings have the same class label c_i as c. Each percentage gives an interpretability score for label c.

The results presented in table 3.4 for `HAMLET-CNN` show that for most classes, the

Figure 3.4: Evolution of confusion matrices for HAMLET−CNN with co-training

Table 3.4: Interpretability scores for HAMLET−CNN.

Class	LRDA	GRDA	LPD	Seizure	GPD
Interpretability	75.00%	68.75%	62.50%	37.50%	25.00%

model is really able to learn how to use the similarity scores. Each percentage shows, out of the 16 reference embeddings with highest weights in the model, how many belong to the target class. Interestingly, the less interpretable classes in the model also are the ones with worst classification performance according to the confusion matrices. One reason could potentially be that the set of patterns that represent such classes is really diverse, preventing the model from efficiently using similarity scores.

Knowing that the model knows how to make use of reference embeddings, the expert re-evaluation task can benefit from interpretability outputs from the model. For a given input sequence that our model suggests as needing re-evaluation, the embeddings with highest similarity scores are selected and displayed next to the sequence, explaining why the model made that particular decision.

Comparison of Machine Feedback Strategies

Finally, we compared the three different strategies introduced in 3.3.7. For each strategy, HAMLET suggested 128 samples with new labels. We counted how many the expert agrees with in each case. The results in table 3.5 show that entropy might be a better indicator of misclassified inputs, although all methods perform almost equally well.

3.5 Conclusion

By acknowledging the difficulty of label acquisition in new domains for both human experts and machine algorithms, we have reached with HAMLET multiple advances in deep

Table 3.5: Expert agreement on suggestions for various machine feedback strategies.

Method	Agreement
High Confidence	92.19%
Margin Sampling	91.41%
Entropy	93.75%

learning for healthcare applications. To summarize, first, we have introduced a novel technique, HAMLET, for human and machine co-learning that is suited for creating high-quality labeled datasets on challenging tasks with a limited budget. This technique has benefits that can appreciated in many deep learning applications. We have shown how this technique applies to the classification of seizures from continuous EEG, a challenging task for both machines and human experts. Finally, we have designed a new kind of network with increased interpretability potential. During the dataset improvement phase, this interpretability via similar reference examples can assist experts re-evaluating sample sequences by explaining the reasons why the algorithm came up with a given output. This work has been presented in (O. Deiss et al., 2018 [46]).

CHAPTER 4
CONCLUSION

In this work, a preliminary study of a standard labeling method has shown that the technique cannot be successfully applied when building a dataset of EEG sequences for the classification of seizures. As presented in this thesis, the multiple issues inherent to the simplicity of the method make it unscalable and unreliable. Therefore, the simple examination and classification of inputs, one after another, is unsuitable in order to obtain high quality datasets for the training of deep neural networks for challenging tasks. In such applications, it cannot be taken for granted that a gold truth label will be successfully assigned to the label. This has been shown extensively in the first chapter of this thesis, with results highlighting the differences of labels among three experts.

A scalable algorithm is a partial solution to this first issue, that first solves the scalability aspect. The cluster-based algorithm allows to obtain a large volume of low-quality labels that are sufficient to train a first classifier. In the second part, this work introduces a robust co-learning framework, `HAMLET`, introducing a general technique for iteratively increasing dataset quality from a low-quality labels. `HAMLET` also includes an interpretable deep neural network with the use of an additional memory module. With `HAMLET`, significant performance improvements were reached, ultimately obtaining a dataset suitable for proper training of the classifier.

Ultimately, the work presented in this thesis can benefit a broader range of domains that share similar labeling challenges. When it comes to labeling large volumes of data for challenging applications of deep learning, it might therefore be interesting to approach the problem in two steps as has been done in this work. First, by obtaining low-quality labels that allow training of a first classifier, then iteratively improving the dataset while easing the understanding by experts of notions and concepts needed for label assignment.

REFERENCES

[1] D. Bender and K. Sartipi, "Hl7 fhir: An agile and restful approach to healthcare information exchange," in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, 2013, pp. 326–331.

[2] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "Smart on fhir: A standards-based, interoperable apps platform for electronic health records," *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 899–908, 2016. eprint: `/oup/backfile/content_public/journal/jamia/23/5/10.1093_jamia_ocv189/2/ocv189.pdf`.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. arXiv: `1512.03385`.

[4] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. arXiv: `1412.5567`.

[5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, 115 EP –, Jan. 2017.

[6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. arXiv: `1609.03499`.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, 529 EP –, Feb. 2015.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, Curran Associates Inc., 2012, pp. 1097–1105.

[10] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," *CoRR*, vol. abs/1702.01923, 2017. arXiv: 1702.01923.

[11] G. V, P. L, C. M, and et al, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016. eprint: /data/journals/jama/935924/joi160132.pdf.

[12] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. B. Westover, M. T. Bianchi, and J. Sun, "SLEEPNET: automated sleep staging system via deep learning," *CoRR*, vol. abs/1707.08262, 2017. arXiv: 1707.08262.

[13] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *Web-Age Information Management*, F. Li, G. Li, S.-w. Hwang, B. Yao, and Z. Zhang, Eds., Cham: Springer International Publishing, 2014, pp. 298–310, ISBN: 978-3-319-08010-9.

[14] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from EEG with deep recurrent-convolutional neural networks," *CoRR*, vol. abs/1511.06448, 2015. arXiv: 1511.06448.

[15] M. M. Zack and R. Kobau, "National and state estimates of the numbers of adults and children with active epilepsy united states, 2015," vol. 66, pp. 821–825, Aug. 2017.

[16] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. eprint: https://doi.org/10.1177/001316446002000104.

[17] N. Gaspard, L. J. Hirsch, S. M. LaRoche, C. D. Hahn, and M. B. Westover, "Interrater agreement for critical care eeg terminology," *Epilepsia*, vol. 55, no. 9, pp. 1366–1373, 2014.

[18] R. R. A, V. J, L. J, and et al, "Association of periodic and rhythmic electroencephalographic patterns with seizures in critically ill patients," *JAMA Neurology*, vol. 74, no. 2, pp. 181–188, 2017.

[19] P. W. Mirowski, Y. LeCun, D. Madhavan, and R. Kuzniecky, "Comparing svm and convolutional networks for epileptic seizure prediction from intracranial eeg," in *2008 IEEE Workshop on Machine Learning for Signal Processing*, Oct. 2008, pp. 244–249.

[20] J. Halford, D Shiau, J. Desrochers, B. Kolls, B. Dean, C. Waters, N. Azar, K. Haas, E Kutluay, G. Martz, *et al.*, "Inter-rater agreement on identification of electrographic

seizures and periodic discharges in icu eeg recordings," *Clinical Neurophysiology*, vol. 126, no. 9, pp. 1661–1669, 2015.

[21] H. A. Haider, R. Esteller, C. D. Hahn, M. B. Westover, J. J. Halford, J. W. Lee, M. M. Shafi, N. Gaspard, S. T. Herman, E. E. Gerard, *et al.*, "Sensitivity of quantitative eeg for seizure identification in the intensive care unit," *Neurology*, vol. 87, no. 9, pp. 935–944, 2016.

[22] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *CoRR*, vol. abs/1701.03551, 2017. arXiv: 1701.03551.

[23] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, 354 EP –, Oct. 2017, Article.

[24] M. H. Alomari, A. Samaha, and K. AlKamha, "Automated classification of L/R hand movement EEG signals using advanced feature extraction and machine learning," *CoRR*, vol. abs/1312.2877, 2013. arXiv: 1312.2877.

[25] A. H. Shoeb and J. V. Guttag, "Application of machine learning to epileptic seizure detection," in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, Aug. 2010, pp. 975–982.

[26] X. Yong and C. Menon, "Eeg classification of different imaginary movements within the same limb," *PLOS ONE*, vol. 10, no. 4, pp. 1–24, Apr. 2015.

[27] F Fürbass, M. Hartmann, J. Halford, J Koren, J Herta, A Gruber, C Baumgartner, and T Kluge, "Automatic detection of rhythmic and periodic patterns in critical care eeg based on american clinical neurophysiology society (acns) standardized terminology," *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 45, no. 3, pp. 203–213, 2015.

[28] J. C. Sackellares, D.-S. Shiau, J. J. Halford, S. M. LaRoche, and K. M. Kelly, "Quantitative eeg analysis for automated detection of nonconvulsive seizures in intensive care units," *Epilepsy & Behavior*, vol. 22, S69–S73, 2011.

[29] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals," *Computers in Biology and Medicine*, 2017.

[30] J. Tan, M. Ung, C. Cheng, and C. Greene, "Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders," vol. 20, pp. 132–43, Jan. 2015.

[31] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., 2016, pp. 2802–2810.

[32] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning – ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 52–59, ISBN: 978-3-642-21735-7.

[33] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98, ACM, 1998, pp. 92–100, ISBN: 1-58113-057-0.

[34] M. Darnstädt, H. U. Simon, and B. Szörényi, "Supervised learning and co-training," in *Algorithmic Learning Theory*, J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 425–439, ISBN: 978-3-642-24412-4.

[35] M. L. Zhang and Z. H. Zhou, "Cotrade: Confident co-training with data editing," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 6, pp. 1612–1626, Dec. 2011.

[36] B. Settles, "Active learning literature survey," Tech. Rep., 2010.

[37] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *CoRR*, vol. abs/1606.04080, 2016. arXiv: 1606.04080.

[38] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," vol. 27, pp. 807–814, Jun. 2010.

[39] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG," *CoRR*, vol. abs/1703.05051, 2017. arXiv: 1703.05051.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. arXiv: 1502.03167.

[41] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015. arXiv: 1511.07289.

[42] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '94, Springer-Verlag New York, Inc., 1994, pp. 3–12, ISBN: 0-387-19889-X.

[43] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, ser. IDA '01, London, UK, UK: Springer-Verlag, 2001, pp. 309–318, ISBN: 3-540-42581-0.

[44] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jun. 1948.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. arXiv: 1412.6980.

[46] O. Deiss, S. Biswal, J. Jin, H. Sun, M. B. Westover, and J. Sun, "HAMLET: Interpretable Human And Machine co-LEarning Technique," *ArXiv e-prints*, Mar. 2018. arXiv: 1803.09702 [cs.AI].